


```

int cpt_inte; // compteur
pour le calcul de notre erreur inetgrale
int parametre; // permet de
corriger si le laser est sorti de la barette

int main()
{

j=0;
cpt=0;
m=0;
parametre=0;

horloge.write(0); // Initialisation du signal horloge à
0;
servo_mot.period_ms(20); // Initialisation période du
servomoteur
servo_mot.pulsewidth_us(1530); // Initialisation en position 0 du
servomoteur

toggle_horloge_ticker.attach(&toggle_horloge, N*300e-9); // interuption
de la boucle while par la fonction toggle_horloge à la frequence de
remplissage

while(1) {

m=0;
cpt_moy=0;
cpt_inte=1;

// calcul dela position maximale:
for(int k = 10; k < 60; k++){
//printf("%d = %d\r\n", k, t[k]);
if (t[k]>m && t[k-1]>0.99*m && t[k+1]>0.99*m){
m=t[k];
position_max=k;
//printf("pōsition=%d\n", position_max);
printf("valeur_max=%d", t[position_max]);
}
erreur_inte=(erreur_inte+32-position_max)/cpt_inte;
cpt_inte=cpt_inte+1;
//printf("erreur=%lf\n", erreur_inte);
}

// calcul de la position moyenne du max sur N boucles

```

```

        if(cpt_moy<N_moy){
            t_pos[cpt_moy]=position_max;
            moy=0;
            for(l=0;l<N_moy;l++) {
                moy=moy+t_pos[l]/N_moy;}
            cpt_moy=cpt_moy+1;
        }
        wait_us(300);
        if(cpt_moy==N_moy){
            cpt_moy=0;
        }

    }
}

void toggle_horloge()

{

// Ecriture de la fonction horloge et lecture du signal ccd
if (cpt!=N){
    if (cpt & 1){ //si impair
        horloge.write(0); // bas horloge creneau
        digital_out.write(0);
        cpt=cpt+1;

        if(j < 64) { t[j]=analog_in.read_u16(); //lecture des tensions
des pixels
            j=j+1; }

    }
    else {
        horloge.write(1); // haut de horloge creneau
        digital_out.write(0);
        cpt=cpt+1;

    }
}

// Declenchement du vidage des 64 pixels :
if(cpt==N){
    digital_out.write(1);
    cpt=0;
    j = 0;
}

// Decalage necssaire pour recentrer le laser au milieu:
delta=32-position_max+32-moy+32-erreur_inte;

```

```
// Seuillage du bruit:
if (t[position_max]>20000){
position_servo=1530-delta*coeff_spatial+parametre;
servo_mot.pulsewidth_us(position_servo);
}

// cas en dehors
else if(t[position_max]>6000){
    if (delta>0){
        position_servo=1530+500;
        servo_mot.pulsewidth_us(position_servo);
        if (t[position_max]>5000){
            parametre=-500; }
    }
    else {
        position_servo=1530-500;
        servo_mot.pulsewidth_us(position_servo);
        if (t[position_max]>5000){
            parametre+=500; }
    }
}

}
```